Chase Joyner

882 Homework 4

November 30, 2016

Problem 1:

1) Consider a generalized linear model framework, where the response variables Y_i have distribution within the exponential family, i.e.

$$f(y_i \mid \theta_i) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{\phi} + c(y_i, \phi)\right\}$$

where the scale parameter ϕ is assumed to be known and θ_i is some function of the regressors and their coefficients which is specified by the data model and the link function. Denote the true mean of the responses as $\mu_i = E[y_i \mid \theta_i]$. The link function g relates the covariates to the mean by

$$g(\mu_i) = \mathbf{X}_i'\boldsymbol{\beta}$$

where $\mathbf{X}_i'$ denotes a p-dimensional vector of covariates for the ith observation. In order perform model fitting, we complete the model specification by placing a normal prior distribution on the regression coefficients $\boldsymbol{\beta}$, i.e. *apriori*

$$\boldsymbol{\beta} \sim N(\mathbf{a}, \mathbf{R}).$$

By specifying the prior and likelihood in this fashion, the posterior distribution is

$$f(\boldsymbol{\beta} \mid \mathbf{Y}, \mathbf{X}) \propto \exp\left\{-\frac{1}{2}(\boldsymbol{\beta} - \mathbf{a})'\mathbf{R}^{-1}(\boldsymbol{\beta} - \mathbf{a}) + \sum_{i=1}^{n}\frac{y_i\theta_i - b(\theta_i)}{\phi}\right\}.$$

The idea is to approximate this posterior distribution with a normal distribution that will be used for the proposal distribution. To achieve this, a second order Taylor expansion of the likelihood term

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n}\frac{y_i\theta_i - b(\theta_i)}{\phi}$$

is carried out around some value of $\boldsymbol{\beta}$, say $\boldsymbol{\beta}^{(t-1)}$, in order to combine with the prior term. The result is a normal distribution with mean vector

$$\mathbf{m}^{(t)} = \left(\mathbf{R}^{-1} + \frac{1}{\phi}\mathbf{X}'\mathbf{W}\big(\boldsymbol{\beta}^{(t-1)}\big)\mathbf{X}\right)^{-1} \times \left(\mathbf{R}^{-1}\mathbf{a} + \frac{1}{\phi}\mathbf{X}'\mathbf{W}\big(\boldsymbol{\beta}^{(t-1)}\big)\widetilde{\mathbf{y}}\big(\boldsymbol{\beta}^{(t-1)}\big)\right)$$

and covariance matrix

$$\mathbf{C}^{(t)} = \left(\mathbf{R}^{-1} + \frac{1}{\phi}\mathbf{X}'\mathbf{W}\big(\boldsymbol{\beta}^{(t-1)}\big)\mathbf{X}\right)^{-1},$$

where $\mathbf{W}\big(\boldsymbol{\beta}^{(t-1)}\big)$ is a diagonal weight matrix with entries

$$W_{ii}\big(\boldsymbol{\beta}^{(t-1)}\big) = \frac{1}{b''(\theta_i)g'(\mu_i)^2}$$

and $\widetilde{\mathbf{y}}\big(\boldsymbol{\beta}^{(t-1)}\big)$ is a vector transformed observations with entries

$$\widetilde{y}_i\big(\boldsymbol{\beta}^{(t-1)}\big) = \mathbf{X}_i'\boldsymbol{\beta} + (y_i - \mu_i)g'(\mu_i).$$

Then, $J_{\boldsymbol{\beta}^{(t)}} = N\big(\mathbf{m}^{(t)}, \mathbf{C}^{(t)}\big)$ approximates the true posterior distribution and thus leads to high acceptance ratios in a metropolis-hastings algorithm. This method is summarized as follows:

1. Initialize $\boldsymbol{\beta}^{(0)}$ and set $t = 1$;
2. Propose $\boldsymbol{\beta}^{\star}$ from $J_{\boldsymbol{\beta}^{(t)}}$ and accept it with probability $p = \min\{r, 1\}$, where

$$r = \frac{f(\boldsymbol{\beta}^{\star} \mid \mathbf{Y}, \mathbf{X})}{f(\boldsymbol{\beta}^{(t)} \mid \mathbf{Y}, \mathbf{X})} \frac{J_{\boldsymbol{\beta}^{\star}}\big(\boldsymbol{\beta}^{(t)}\big)}{J_{\boldsymbol{\beta}^{(t)}}\big(\boldsymbol{\beta}^{\star}\big)};$$

3. Increment t and return to step 2.

The construction of the proposal parameters $\mathbf{m}^{(t)}$ and $\mathbf{C}^{(t)}$ approximates the posterior mode and covariance matrix for $\boldsymbol{\beta}$. Now, we describe this approach under a logistic regression model.

2) Consider the following logistic data model

$$y_i \sim Bin(\pi_i, 1)$$
$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 x_i.$$

First, the distribution of the observed data is a member of the exponential family since

$$f(y_i \mid \pi_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i} = \left(\frac{\pi}{1-\pi_i}\right)^{y_i}(1-\pi_i)$$
$$= \exp\left\{y_i \log\frac{\pi_i}{1-\pi_i} + \log(1-\pi_i)\right\}.$$

Therefore, we see that $\theta_i = \log\dfrac{\pi_i}{1-\pi_i}$, $b(\theta_i) = \log(1+e^{\theta_i})$, and $\phi = 1$. For notational ease,

$$\boldsymbol{\beta} = (\beta_0, \beta_1)', \quad \mathbf{Y} = (y_1, ..., y_n)', \quad \text{and} \quad \mathbf{X} = (\mathbf{1}_n, (x_1, ..., x_n)').$$

Placing a normal prior distribution on $\boldsymbol{\beta}$, the posterior distribution is

$$f(\boldsymbol{\beta} \mid \mathbf{Y}, \mathbf{X}) \propto \prod_{i=1}^{n} f(y_i \mid \pi_i) f(\boldsymbol{\beta}) \propto \exp\left\{-\frac{1}{2}(\boldsymbol{\beta}-\mathbf{a})'\mathbf{R}^{-1}(\boldsymbol{\beta}-\mathbf{a}) + \sum_{i=1}^{n} y_i \log\frac{\pi_i}{1-\pi_i} + \sum_{i=1}^{n}\log(1-\pi_i)\right\},$$

where π_i is obtained by inverting the link function, i.e.

$$\pi_i = \frac{\exp\{\beta_0 + \beta_1 x_i\}}{1 + \exp\{\beta_0 + \beta_1 x_i\}}.$$

Now we compute the necessary components for the proposal distribution, namely

$$\begin{aligned} b''(\theta_i) &= \frac{e^{\theta_i}}{\left(1 + e^{\theta_i}\right)^2} \\ g'(\mu_i) &= \frac{1}{\pi_i(1 - \pi_i)} \qquad\qquad \text{since } \mu_i = \pi_i. \end{aligned}$$

It now follows that the diagonal weight matrix $\mathbf{W}\left(\boldsymbol{\beta}^{(t-1)}\right)$ has entries

$$W_{ii}\left(\boldsymbol{\beta}^{(t-1)}\right) = \pi_i(1 - \pi_i)$$

and the transformed observations take the form

$$\widetilde{y}_i\left(\boldsymbol{\beta}^{(t-1)}\right) = \beta_0 + \beta_1 x_i + \frac{y_i - \pi_i}{\pi_i(1 - \pi_i)}.$$

Finally, the proposal distribution in the metropolis-hastings algorithm is $N(\mathbf{m}^{(t)}, \mathbf{C}^{(t)})$, where

$$\begin{aligned} \mathbf{m}^{(t)} &= \mathbf{C}^{(t)} \times \left(\mathbf{R}^{-1}\mathbf{a} + \mathbf{X}'\mathbf{W}\left(\boldsymbol{\beta}^{(t-1)}\right)\widetilde{\mathbf{y}}\left(\boldsymbol{\beta}^{(t-1)}\right)\right) \\ \mathbf{C}^{(t)} &= \left(\mathbf{R}^{-1} + \mathbf{X}'\mathbf{W}\left(\boldsymbol{\beta}^{(t-1)}\right)\mathbf{X}\right)^{-1}. \end{aligned}$$

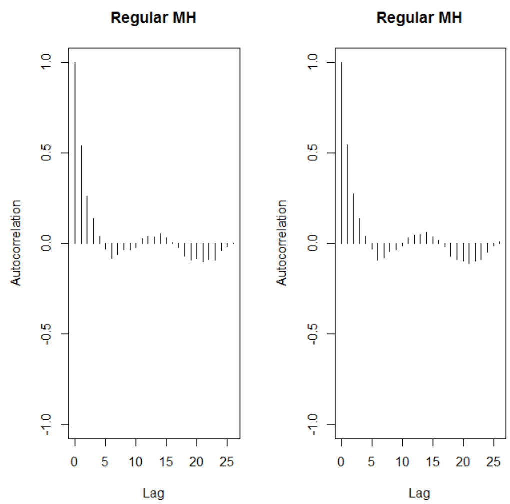
Now the full algorithm summarized in part 1 of this problem can be implemented.

3) The code can be found in the appendix.

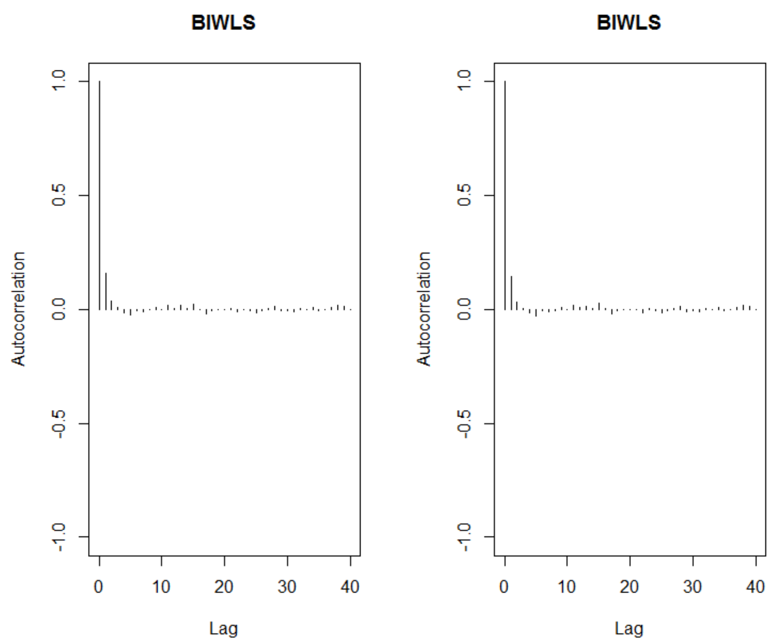
4) The estimates from both algorithms can be found below:

Parameter	BIWLS	Regular MH
β_0	-4.055	-4.031
β_1	0.100	0.099

Therefore, both algorithms seem to approximate the parameters in a similar fashion. We will analyze these two approaches further. Firstly, the autocorrelation using a regular MH algorithm is:



which is the autocorrelation plot for β_0 and β_1, respectively. The autocorrelation plot using the BIWLS algorithm is



for β_0 and β_1, respectively. The autocorrelation for the BIWLS is far less, and this was even without any thinning. Lastly, the effective sample size for β_0 was 7270 out of 10000 samples and for β_1 was 7898 out of 10000 samples with the BIWLS. Comparatively, the effective sample size was 311 for β_0 and 305 for β_1 out of 1000 samples, which also is not as good as the BIWLS method. The last note worthy thing to mention is the acceptance rate for (β_0, β_1) was about 95% while the acceptance rate for β_0 and β_1 under the regular MH algorithm was about 24% each.

Problem 2:

1) The idea of a conditional means prior (CMP) is with p predictors, i.e. p parameters, we include our prior information to construct an informative prior distribution to be used in a generalized linear model framework. This is typical tough to do in GLMs. In a GLM, we have

$$m_i := E[Y_i \mid \mathbf{X}_i] = g^{-1}(\mathbf{X}_i'\boldsymbol{\beta})$$

where g is a specified link function that relates the covariates to the mean response. The informative prior for $\boldsymbol{\beta}$ is induced by a CMP on the new mean vector $\widetilde{\mathbf{m}} = (\widetilde{m}_1, ..., \widetilde{m}_n)$, where

$$\widetilde{m}_i = E[\widetilde{Y}_i \mid \widetilde{\mathbf{X}}_i]$$

where $\widetilde{Y}_i$ is some observable response at covariate $\widetilde{\mathbf{X}}_i$, where the selected p covariate vectors $\widetilde{\mathbf{X}}_i$ are linearly independent. Now, for each new mean $\widetilde{m}_i$, elicit the independent prior $\pi(\widetilde{m}_i)$. Then, the CMP is given by

$$\pi(\widetilde{\mathbf{m}}) = \prod_{i=1}^{p} \pi(\widetilde{m}_i).$$

Define $\mathbf{G}(\widetilde{\mathbf{m}})$ that applies g to each element, i.e. entries $g(\widetilde{m}_i)$. The link function gives the relationship $\boldsymbol{\beta} = \widetilde{\mathbf{X}}^{-1}\mathbf{G}(\widetilde{\mathbf{m}})$, and so through a transformation the induced prior is

$$\begin{aligned}\pi(\boldsymbol{\beta}) &\propto \prod_{i=1}^{p} \pi(\widetilde{m}_i)/g'\left(\widetilde{m}_i\right) \\ &\propto \prod_{i=1}^{p} \pi\left(g^{-1}(\widetilde{\mathbf{X}}_i\boldsymbol{\beta})\right)/g'\left(g^{-1}(\widetilde{\mathbf{X}}_i\boldsymbol{\beta})\right).\end{aligned}$$

2) Now we outline the details to develop the CMP for a logistic regression model. Consider the model of the form

$$\begin{aligned} Y_i &\sim Bin(m_i, 1) \\ g(m_i) &= \log\left(\frac{m_i}{1-m_i}\right) = \mathbf{X}_i\boldsymbol{\beta}\end{aligned}$$

where

$$m_i = E[Y_i \mid \mathbf{X}_i] = g^{-1}(\mathbf{X}_i\boldsymbol{\beta}) = \frac{\exp\{\mathbf{X}_i\boldsymbol{\beta}\}}{1+\exp\{\mathbf{X}_i\boldsymbol{\beta}\}}$$

which is a logistic regression model. Notice that

$$g'(m_i) = m_i^{-1}(1-m_i)^{-1} \quad \implies \quad g'\left(g^{-1}(\widetilde{\mathbf{X}}_i\boldsymbol{\beta})\right) = \frac{\left(1+\exp\{\mathbf{X}_i\boldsymbol{\beta}\}\right)^2}{\exp\{\mathbf{X}_i\boldsymbol{\beta}\}}.$$

Then, once specifying the independent priors $\pi(\widetilde{m}_i)$, the induced prior is

$$\pi(\boldsymbol{\beta}) \propto \prod_{i=1}^{p} \pi\left(g^{-1}(\widetilde{\mathbf{X}}_i\boldsymbol{\beta})\right)\frac{\exp\{\mathbf{X}_i\boldsymbol{\beta}\}}{\left(1+\exp\{\mathbf{X}_i\boldsymbol{\beta}\}\right)^2}.$$

In logistic regression, a reasonable choice for the independent priors are

$$\widetilde{m}_i \sim \text{Beta}(a_i, b_i)$$

and therefore the induced prior for the regression coefficients is

$$\begin{aligned}
\pi(\boldsymbol{\beta}) &\propto \prod_{i=1}^{p} \left(g^{-1}(\widetilde{\mathbf{X}}_i\boldsymbol{\beta})\right)^{a_i-1} \left(1 - g^{-1}(\widetilde{\mathbf{X}}_i\boldsymbol{\beta})\right)^{b_i-1} \frac{\exp\{\mathbf{X}_i\boldsymbol{\beta}\}}{\left(1+\exp\{\mathbf{X}_i\boldsymbol{\beta}\}\right)^2} \\
&\propto \prod_{i=1}^{p} \left(g^{-1}(\widetilde{\mathbf{X}}_i\boldsymbol{\beta})\right)^{a_i} \left(1 - g^{-1}(\widetilde{\mathbf{X}}_i\boldsymbol{\beta})\right)^{b_i}.
\end{aligned}$$

3) Now we develop a sampling strategy to draw posterior samples for $\boldsymbol{\beta}$ under the CMP prior using the technique discussed in problem 1. Consider a binomial data model of the form

$$\begin{aligned}
\widetilde{Y}_i &\sim \text{Bin}(n_i, p_i) \\
\log\left(\frac{p_i}{1-p_i}\right) &= \widetilde{\mathbf{X}}_i\boldsymbol{\beta}.
\end{aligned}$$

The likelihood function for a single observation in this situation is given by

$$f\left(\widetilde{Y}_i \mid \widetilde{\mathbf{X}}_i\right) \propto p_i^{Y_i}(1-p_i)^{n_i - Y_i}.$$

Comparing this function to the induced prior in part 2), we define the parameters in the following way:

$$a_i = \widetilde{Y}_i \quad \text{and} \quad b_i = n_i - \widetilde{Y}_i.$$

Now, for sampling, we implement the method discussed in problem 1. Combine variables as

$$\begin{aligned}
\mathbf{Y}^\star &= (\mathbf{Y}, \widetilde{Y}_1, ..., \widetilde{Y}_p)' \\
\mathbf{X}^\star &= (\mathbf{X}, \widetilde{\mathbf{X}}_1, ..., \widetilde{\mathbf{X}}_p)'
\end{aligned}$$

where $\mathbf{Y} = (Y_1, ..., Y_n)$ defined in part 2). Then, using the derivations from problem 1,

$$\begin{aligned}
\mathbf{C} &= \left(\mathbf{X}^{\star T}\mathbf{W}(\boldsymbol{\beta})\mathbf{X}^\star\right)^{-1} \\
\mathbf{m} &= \mathbf{C} \times \left(\mathbf{X}^{\star T}\mathbf{W}(\boldsymbol{\beta})\overline{\mathbf{Y}}^\star(\boldsymbol{\beta})\right)
\end{aligned}$$

where $\overline{\mathbf{Y}}^\star(\boldsymbol{\beta})$ is a vector of transformed observations, i.e. transformation of $\mathbf{Y}^\star$, with entries

$$\begin{aligned}
\overline{Y}_j^\star(\boldsymbol{\beta}) &= \mathbf{X}_j\boldsymbol{\beta} + \frac{Y_j - m_j}{m_j(1-m_j)} \quad \text{for } j = 1, ..., n \\
\overline{Y}_i^\star(\boldsymbol{\beta}) &= \widetilde{\mathbf{X}}_i\boldsymbol{\beta} + \frac{\widetilde{Y}_i - n_i p_i}{n_i p_i(1-p_i)} \quad \text{for } i = 1, ..., p
\end{aligned}$$

and $\mathbf{W}(\boldsymbol{\beta})$ is the diagonal weight matrix with entries

$$W_{ii}(\boldsymbol{\beta}) = n_i p_i (1-p_i).$$

Then, we use the algorithm discussed in problem 1.

Problem 3:

Here we develop a metropolis algorithm to sample the ρ term in the CAR model from problem 2 of homework 3. Namely, consider the spatial regression model given by

$$Y_i = \beta_0 + b_i + \epsilon_i,$$

where $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$ for $i = 1, ..., n$. We assume that the spatial random effects follow the CAR$(\sigma^2\tau^2, \rho)$ model, i.e.

$$\mathbf{b} = (b_1, ..., b_n)' \sim N(\mathbf{0}, \sigma^2\tau^2(\mathbf{D} - \rho\mathbf{W}).$$

It can be shown that ρ must be between -1 and 1, and as a result we will assume $\rho \sim \text{Unif}(-1, 1)$. Then, the posterior distribution for ρ is simply

$$\pi(\rho \mid \mathbf{b}, \sigma^2, \tau^2) \propto \exp\left\{-\frac{1}{2\sigma^2\tau^2}\mathbf{b}'(\mathbf{D} - \rho\mathbf{W})\mathbf{b}\right\}.$$

Now, for the proposal distribution, we will consider a symmetric random walk, namely

$$J\big(\rho \mid \rho^{(t)}\big) = \text{Unif}\big(\rho^{(t)} - c, \rho^{(t)} + c\big)$$

where c is a tuning parameter to achieve an acceptance probability of roughly 35%. Denote $\boldsymbol{\theta}_{-\rho}$ as the set of parameters to update, excluding ρ. Then, the algorithm is as follows:

1. Initialize $\boldsymbol{\theta}^{(0)}$ and set $t = 1$.
2. Update $\boldsymbol{\theta}_{-\rho}^{(t)}$ via the Gibbs sampling scheme of problem 2 of homework 3.
3. Propose $\rho^\star$ from $J\big(\rho \mid \rho^{(t)}\big)$ and accept it with probability $p = \min\{r, 1\}$, where

$$r = \frac{\pi(\rho^\star \mid \mathbf{b}, \sigma^2, \tau^2)}{\pi(\rho^{(t)} \mid \mathbf{b}, \sigma^2, \tau^2)}$$

APPENDIX

```
####################################
####################################
#####
#####                  Chase Joyner
#####               882 Homework 4
#####
####################################
####################################

## Required libraries ##
library(MASS)
library(mvtnorm)
library(coda)

####################################
##### Problem 1

BIWLS = function(Y, X, iter = 1e4){

        ## Preliminaries ##
        n = length(Y)
        p = dim(X)[2]
        R = 100*diag(p)
        a = rep(1, p)

        ## Initial values ##
        beta = rep(0, p)
        acc = 0

        ## Save records ##
        Beta = matrix(-99, nrow = iter, ncol = p)

        ## Compute only once ##
        IR = solve(R)
        IRa = IR %*% a

        ## Link function for pi ##
        link = function(u){
                expXB = exp(X %*% u)
                val = expXB / (1 + expXB)
                return(val)
        }

        llik = function(s){
```

```
                pi = link(s)
                val = -(1/2)*t(s-a)%*%IR%*%(s-a) + sum(Y*log(pi/(1-pi))) + sum(log(
                return(val)
        }

        for(i in 1:iter){
                ## Update beta ##
                pi.t = as.vector(link(beta))
                newY.t = X %*% beta + (Y - pi.t) / (pi.t * (1 - pi.t))
                Wt = diag(pi.t * (1 - pi.t))
                Ct = solve(IR + t(X) %*% Wt %*% X)
                mt = Ct %*% (IRa + t(X) %*% Wt %*% newY.t)
                beta.s = as.vector(rmvnorm(1, mt, Ct))
                pi.s = as.vector(link(beta.s))
                newY.s = X %*% beta.s + (Y - pi.s) / (pi.s * (1 - pi.s))
                Ws = diag(pi.s * (1 - pi.s))
                Cs = solve(IR + t(X) %*% Ws %*% X)
                ms = Cs %*% (IRa + t(X) %*% Ws %*% newY.s)
                r = exp(llik(beta.s) - llik(beta) + dmvnorm(beta, ms, Cs, log = TRU
                z = rbinom(1, 1, min(r, 1))
                if(z == 1){
                        beta = beta.s
                        acc = acc + 1
                }
                Beta[i,] = beta
                print(c(i))
        }
        return(list(Beta = Beta, accept = acc / iter))
}

## Generate some data and run ##
#par(mfrow = c(2,1))
#n = 1000
#beta.true = c(-1, 0.5)
#X = cbind(rep(1, n), rnorm(n, 2, 1))
#probs = exp(X %*% beta.true) / (1 + exp(X %*% beta.true))
#Y = rbinom(n, 1, probs)
#res = BIWLS(Y, X)
#apply(res$Beta, 1, mean)

## Analyze diabetes data ##
library(MASS)
data(Pima.tr)
data(Pima.te)
pima = rbind(Pima.tr, Pima.te)
```

```
Y = as.vector(pima[, 8])
Y[Y == "Yes"] = 1
Y[Y == 'No'] = 0
Y = as.numeric(Y)
X = cbind(1, as.matrix(pima[,5]))
res = BIWLS(Y, X)
apply(res$Beta, 2, mean)
Beta.mcmc1 = as.mcmc(res$Beta[1,])
Beta.mcmc2 = as.mcmc(res$Beta[2,])
autocorr.plot(Beta.mcmc1)
autocorr.plot(Beta.mcmc2)
effectiveSize(Beta.mcmc1)
effectiveSize(Beta.mcmc2)
```